

REMARKS

This responds to the Non-Final Office Action dated 2 September 2008. No new matter has been added. Claims 1, 3-8, and 10-15 are presently pending in the application, each of which Applicant believes is in condition for allowance. Applicant respectfully requests reconsideration in light of the above amendments and the following remarks.

For simplicity and clarity purposes in responding to the Office Action, Applicant's remarks are primarily focused on the rejections applied to the independent claims (*i.e.*, claims 1 and 8) as outlined in the Office Action with the understanding that the dependent claims are patentable for at least the same reasons (and in most cases other reasons) that the independent claims are patentable. Applicant expressly reserves the right to argue the patentability of the dependent claims separately in any future proceedings.

Substance of the Interview

Applicant thanks the Examiner for discussing the application with Applicant's representatives, Bryan K. Hanks and Christopher J. Wickstrom, on 15 October 2008. Claim 1 and the Hall, Curtis, and Krishnan references were discussed. Although no agreement was reached, Applicant appreciates the Examiner's thoughts and insights with respect to the present Application.

Claim Rejections – 35 U.S.C. § 103

In the Action, Examiner rejected claims 1, 3-8, and 10-15 under 35 U.S.C. § 103(a) as allegedly being unpatentable over pp. 495-502 of a publication to Hall et al. entitled "A Virtual Operating System" ("Hall") in view of U.S. Patent No. 6,615,277 to Curtis ("Curtis"), and further in view of U.S. Patent No. 6,141,698 to Krishnan et al.

("Krishnan"). Applicant traverses these rejections for at least the reasons set forth below.

A. Claims 1 and 8

For at least the reasons discussed below, Hall, Krishnan, and Curtis, taken either alone or in combination, fail to support a 35 U.S.C. § 103 rejection of claims 1 and 8. The references cited by Examiner do not show, teach, or suggest various features of the claims. Furthermore, Examiner has not provided any evidence, or even a logical assertion, that suggests any motivation to combine the cited references to provide the features of the claims.

According to Federal Circuit precedent, the burden of establishing a *prima facie* case of obviousness under 35 U.S.C. § 103 rests squarely on the shoulders of the examiner. To establish a *prima facie* case of obviousness, the reference (or references when combined) must teach or suggest each and every claim element. *See, e.g., In re Royka*, 490 F.2d 981, 985 (CCPA 1974); *accord*. MPEP 2143.03. Indeed, as the Board of Patent Appeal and Interferences has recently confirmed in a 2007 decision, a proper obviousness determination requires that an Examiner make "a searching comparison of the claimed invention – *including all its limitations* – with the teaching of the prior art." *See In re Wada and Murphy*, Appeal 2007-3733, *citing In re Ochiai*, 71 F.3d 1565, 1572 (Fed. Cir. 1995) (emphasis added). As shown below, Examiner has not shown how the references teach various claim elements.

Additionally, it is established law that one "cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention." *Ecolochem, Inc. v. Southern Cal. Edison Co.*, 227 F.3d 1361, 1371, 56 USPQ2d 1065 (Fed. Cir. 2000) (citing *In re Fine*, 837 F.2d 1071, 1075, 5 USPQ2d

1780, 1783 (Fed. Cir. 1988)). Indeed, “[c]ombining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor’s disclosure as a blueprint for piecing together the prior art to defeat patentability – the essence of hindsight.” *In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999). During the interview, Examiner often referred to the present application, rather than any prior references, to attempt to piece together unrelated references to attempt to show how the references could be combined to make the pending claims obvious.

1. The cited references do not teach a “virtual OS environment having a ... virtual OS registry” that is “independent of the base OS file system and the base OS registry”

As admitted in the Office Action, Hall and Krishnan fail to disclose an operating system having an “OS registry” as recited in claims 1 and 8. Similarly, Curtis fails to show, teach, or suggest a “virtual OS environment having a . . . virtual OS registry” that is “independent of the base OS file system and the base OS registry” as recited in claim 1, or a “virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry” as recited in claim 8. Rather, Curtis discloses a global registry object that is used to map registry functions in multiple non-virtual OS environments in order to simplify the job of a programmer writing install code for each of the multiple non-virtual OS environments. *See, e.g.*, Curtis, col. 4, lines 17-44 and col. 6, lines 5-28.

For example, Curtis states:

It would also be desirable for a software manufacturer to have a common look and feel for writing install code for all of its products. In this way, as a programmer moved from platform to platform in writing install code, the

programmer would recognize the interface, and know how it works.

Curtis, col. 4, lines 38-43. In order to overcome the above issue of variation between platforms (*i.e.*, operating systems), Curtis uses a global registry object to make the interface between the operating systems and the programmer more consistent. *See, e.g.*, Curtis, col. 6, lines 5-28. As explained in Curtis:

[T]here is a global registry object for carrying out, *i.e.*, mapping, registry functions or registry equivalent functions across multiple operating systems. A developer is enabled to create a platform independent program that can read, create, modify, delete, and enumerate registry type of information regardless of whether or not a targeted operating system supports a registry or registry equivalent functionality.

Curtis, col. 6, lines 6-14 (emphasis added). Curtis makes clear that that the global registry object is designed to work across different operating systems at different times. In other words, the registry object enables an operator to maintain a familiar interface regardless of which OS environment the operator is using (*i.e.*, Windows, OS/2, AIX, etc.). Basically, the global registry object in Curtis enables an operator to easily transition from using or programming a particular program, such as an install program, in different operating system environments, without having to use a different interface in each of the different operating system environments.

However, while Curtis arguably describes base OS environments having a registry, as well as a global registry object that may carry out various functions in different types of OS environments, Curtis fails to show, teach, or even suggest a “virtual OS environment having a . . . virtual OS registry” that is “independent of the base OS file system and the base OS registry” as recited in claim 1, or a “virtual OS

environment having a virtual file system and registry which are independent of the base OS file system and registry” as recited in claim 8.

In fact, Curtis fails to show, teach, or even suggest a virtual OS environment at all. Rather, Curtis merely describes base OS environments. Accordingly, Curtis necessarily fails to teach a virtual registry or a virtual OS registry that might be located in a virtual OS environment. Curtis therefore fails to cure the deficiencies of Hall and Krishnan described above. Accordingly, Applicant respectfully requests that the rejection of independent claims 1 and 8 be withdrawn.

2. The cited references fail to teach “[A]ttempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry”

Hall and Krishnan fail to show, teach, or suggest that “attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry” as recited in claim 1. Hall and Krishnan also fail to teach “attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry” as recited in claim 8 (emphasis added).

On page 5 of the Action, Examiner concedes that Hall does not disclose the aforementioned feature of independent claim 1. Examiner rejects independent claims 1 and 8, contending that the secondary citation to Krishnan provides the necessary disclosure.

In rejecting independent claims 1 and 8, Examiner relies on Krishnan and contends that Krishnan “teaches a change made in the OS (inject dll, abstract, figures 2,

3, and 6), wherein the change does not affect the main OS or any other virtual OS environment (by inject dll not modifying the current code of any operating systems)."

Office Action at page 4. Examiner makes the following conclusory statement at page 4 of the Office Action:

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Hall, Curtis, and Krishma's system because Krishma's injecting dll to the virtual OS would run the code in the dll that change the execution to redirect to the virtual OS system, and no need to modify the code of the operating system, and therefore, it the maintenance of application is easier.

Applicant respectfully disagrees with Examiner's conclusions, finding no evidence or suggestion of the aforementioned claimed features in Krishnan. Further, there is no suggestion to even combine the teachings, as advanced in the Action, except from using Applicant's invention as a template through hindsight reconstruction of Applicant's claims.

In contrast to Examiner's assertion, Krishnan teaches:

A method and system for modifying the behavior of existing executable code by injecting new code into an executable file is provided. The injection mechanism injects a reference to new code contained in a DLL into an existing executable file such that, when the code of the executable file is executed, the DLL is automatically loaded and the new code is automatically executed. A reference to the DLL is injected into the executable file by either modifying an import table of the file, which causes automatic loading of the DLLs referred to therein, or by adding DLL loader code to the file.

Abstract. As can be seen, Krishnan teaches that injecting code into an executable file may be helpful in situations such as when a third party vendor, who does not have access to application source code, “may wish to incorporate vendor-specific code into the application before redistributing to an end customer.” Krishnan, col. 1, lines 23-30. As another example, Krishnan teaches that injecting new code into existing application code is helpful in preventing “software pirates” from decrypting application programs and making illegal copies of such application programs. *See, e.g.*, Krishnan, col. 1, lines 31 to col. 2, line 28).

Krishnan additionally teaches that “injection of new code enables the existing executable code to perform new behaviors. For example, licensing procedures can be added to an existing application by injecting a licensing DLL into the application using the injection mechanism” *See, e.g.*, Krishnan, col. 2, lines 37-41. In other words, the injected DLL of Krishnan redirects a user to new code that enables existing code to perform new behaviors, such as presenting licensing procedures to a user to prevent software piracy. This simple modification to existing code using a DLL, as disclosed in Krishnan, is entirely different from redirecting “attempts to access [a] base OS file system and [a] base OS registry” to a “virtual OS file system and [a] virtual OS registry” according to the recitations in claims 1 and 8.

Clearly, Krishnan fails to show, teach, or even suggest, either alone or in combination with Hall and Curtis, that “attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry” as recited in claim 1 or that “attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS

environment file system and registry” as recited in claim 8. Accordingly, Applicant respectfully requests that the rejection of independent claims 1 and 8 be withdrawn.

3. The cited references fail to teach “[T]he virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry”

Hall does not disclose a “virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry” as is recited in claim 1 or “each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry” as is recited in claim 8.

Hall is an old and dated citation that is alleged to have been published in 1980. In this reference, Hall attempts to solve a problem that existed back in this era, namely, a problem of organizations being hesitant to move to new hardware systems (with their associated operating systems) because of the high costs associated with having to train personnel on the new systems and porting software over to the new operating systems. *See, e.g.,* Hall, page 495, cols. 1-3. Hall proposes a “uniform system interface.” Hall, page 495, col. 1 and col. 3. According to Hall, this is effective because with computer users “there is no need to distinguish between the interface to an operating system and the operating system itself.” Hall, page 495, col. 3. FIG. 1 of Hall further shows how this virtual machine consists of “interfacing the standardized virtual machine to the vendor supplied system.” In essence, this virtual machine is simply a wrapper on top of a base operating system.

In rejecting these claim elements, Examiner alleges that Hall, at page 497, col. 1, section 4, discloses “file systems.” The section of Hall cited in the Action states:

“To test the approach, a uniform program development environment consists of resources which assist programmers in the development and maintenance of computer programs, such as text editors, programming language processors, and file systems. The types of system resources with which such a virtual machine is concerned (files, directories, processes, and the user environment) require a general-purpose operating system interface.”

Hall, page 497, col. 1, section 4. The reference to “file systems” in this section of Hall does not include any teaching or suggestion that the “file systems” are part of a virtual OS environment or that the file systems are independent of a base OS file system. Rather, the reference merely describes resources (*e.g.*, a text editor) that are available in a program development environment to assist programmers in the development and maintenance of computer programs. Moreover, the program development environment in the cited section of Hall is used merely to create source code for an operating system, and there is no teaching or suggestion in Hall of the operation system having a file system that is independent of the base OS file system.

Examiner also cites to a portion of page 497 of Hall, which states:

In all cases, the system was offered in parallel with the existing environment, thereby allowing users to experiment with the virtual operating system without giving up the familiar, vendor-supplied environment.

Page 497, col. 1, section 4 (emphasis added). Examiner uses this citation in an attempt to show the virtual operating system of Hall as being independent of the base OS file system and the base OS registry. However, Applicant respectfully disagrees with this argument. The reference to “parallel” does not mean that the virtual operating system (*i.e.*, system interface) is “independent of the base OS file system and the base OS

registry” as is asserted in the Action. Instead, the term “parallel” means simultaneously or concurrently. In other words, the virtual operating system (*i.e.*, system interface) may be run simultaneously or concurrently with the “familiar vendor-supplied” interface of the base operating system.

In fact, as discussed above, Hall teaches away from a “virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry” as is recited in claim 1, or “each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry” as is recited in claim 8. For instance, the disclosed purpose of the virtual operating system in Hall is to provide a standard interface to a real operating system. *See, e.g.*, Hall, page 496, FIG. 1 and col. 1, first and second paragraphs.

In other words, the virtual operating system in Hall is simply a wrapper on top of a real operating system, *i.e.*, an interface between a user interface and a real operating system. Hall states, “the emphasis in building a virtual operating system is on the interface presented to the user.” Hall, page 496, col. 1, fourth paragraph. As a mere interface on top of a real operating system, there is no need for the virtual operating system of Hall to have a file system or a registry that is independent of the file system or registry of the real operating system. In fact, by focusing on providing a standardized interface to a real operating system, Hall teaches away from such a configuration. *See, e.g.*, Hall, page 495, col. 3, last paragraph).

According, Hall fails to disclose a “virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry” as is recited in claim 1 or “each virtual OS environment

having a virtual file system and registry which are independent of the base OS file system and registry” as is recited in claim 8 (emphasis added).

Krishnan and Curtis are cited for their alleged disclosure of various features of claims 1 and 8 other than the aforementioned features. Applicant respectfully submits that neither Krishnan nor Curtis adds anything to the disclosure of Hall that would remedy the aforementioned deficiencies. Accordingly, for at least the aforementioned reasons, Applicant respectfully requests that the rejection of independent claims 1 and 8 be withdrawn.

4. The cited references fail to teach “[A]t least one virtual OS environment within the base OS”

Additionally, Hall, Krishnan, and Curtis also fail to disclose “at least one virtual OS environment within the base OS” as recited in claim 1 or “creating at least one virtual OS environment under the base OS” as recited in claim 8.

In fact, Hall and Krishnan teach away from these claim elements. For example, FIG. 1 of Hall shows that the virtual operating system, as an interface to a real operating system, is located on top of a vendor supplied system (*i.e.*, the virtual operating system of Hall is a wrapper for a real operating system). Therefore, Hall fails to disclose “at least one virtual OS environment within the base OS” as recited in claim 1, or “creating at least one virtual OS environment under the base OS” as recited in claim 8 (emphasis added). Accordingly, Applicant respectfully requests that the rejection of independent claims 1 and 8 be withdrawn.

B. Claims 3-7, 10, and 11

Claims 3-7, 10, and 11 depend from independent claims 1 and 8. By virtue of this dependency, Applicant submits that claims 3-7, 10, and 11 are allowable for at least

the same reasons given above with respect to claims 1 and 8. In addition, Applicant submits that claims 3-7, 10, and 11 are further distinguished over cited art by the additional elements recited therein, and particularly with respect to each claimed combination. Applicant respectfully requests, therefore, that the rejection of claims 3-7, 10, and 11 under 35 U.S.C. § 103 be withdrawn, and these claims be allowed.

C. Claims 12 and 13

Claims 12 and 13 depend from independent claim 8. By virtue of this dependency, Applicant submits that claims 12 and 13 are allowable for at least the same reasons given above with respect to claim 8. Claims 12 and 13 also recite additional subject matter not disclosed in Hall, Krishnan, or Curtis. For example, claim 12 recites “creating a copy of the base OS file system and registry in the virtual OS environment file system and registry.” Additionally, claim 13 recites “wherein the application running under the virtual OS environment is executed using the copy in the virtual OS environment file system and registry.”

Examiner asserted that Krishnan teaches the above feature of claims 12 and 13 at column 4, line 55 to column 5, line 5. *See* Office Action at page 6. Applicant respectfully submits that the above citation from Krishnan is silent concerning “creating a copy of the base OS file system and registry in the virtual OS environment file system and registry” or that an “application running under the virtual OS environment is executed using the copy in the virtual OS environment file system and registry.” Rather, the above citation from Krishnan merely teaches using an “injection mechanism ... to create a modified version of the application that includes a reference to the new DLL.” Krishnan, col. 4, lines 63-66. Applicant respectfully requests, therefore, that the rejection of claims 12 and 13 under 35 U.S.C. § 103 be withdrawn.

D. Claim 14

Claim 14 depends from independent claim 8. By virtue of this dependency, Applicant submits that claim 14 is allowable for at least the same reasons given above with respect to claim 8. In addition, claim 14 recites subject matter not disclosed in Hall, Krishnan, or Curtis. For example, claim 14 recites “setting a predetermined directory such that an application running under the predetermined directory will be redirected to the virtual OS environment based on the location of the application being under the predetermined directory.”

Regarding claim 14, Examiner contends on page 7 of the Action that “it would have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that the injecting DLL method and techniques can be applied to different computing environment because the injecting DLL is portable and convenient to attach anywhere in the application as designed to redirect the execution without being modified the existing code.”

Examiner's above argument is unclear as it does not address the features recited in claim 14, nor does the Examiner show where in Hall, Krishnan, or Curtis these features may be found. Applicant has carefully reviewed the references cited in the Action, and submits that neither Hall, Krishnan, nor Curtis teaches “setting a predetermined directory such that an application running under the predetermined directory will be redirected to the virtual OS environment based on the location of the application being under the predetermined directory.” Accordingly, for at least these reasons, the rejection of claim 14 under 35 U.S.C. § 103 should be withdrawn.

E. Claim 15

Claim 15 depends from claims 8 and 14. By virtue of this dependency, Applicant submits that claim 15 is allowable for at least the same reasons given above with respect to claims 8 and 14. In addition, Applicant submits that claim 15 is further distinguished over cited art by the additional elements recited therein, and particularly with respect to each claimed combination. Applicant respectfully requests, therefore, that the rejection of claim 15 under 35 U.S.C. § 103 be withdrawn, and this claim be allowed.

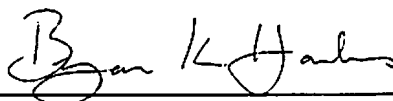
Conclusion

For at least the foregoing reasons, Applicant believes that each of the presently pending claims in this application is in immediate condition for allowance. Accordingly, Applicant respectfully requests a favorable action on the merits. If Examiner has any further comments or suggestions, Applicant invites Examiner to contact the undersigned attorney to expedite the handling of this matter.

Applicant expressly disclaims all arguments, representations, and/or amendments presented or contained in any other patent or patent application, including any patents or patent applications claimed for priority purposes by the present application or any patents or patent applications that claim priority to this patent application. Moreover, all arguments, representations, and/or amendments presented or contained in the present patent application are only applicable to the present patent application and should not be considered when evaluating any other patent or patent application.

Respectfully submitted,

Date: 20 October 2008



Bryan K. Hanks
Registration No. 52,991